

REMARKS

Applicants submitted an Information Disclosure Statement (IDS) with the filed application on April 25, 2000. The Examiner did not indicate that he reviewed the IDS. Applicants submit herewith a copy of the IDS submitted with the application and a copy of the Return Postcard submitted with the Application showing the Patent Office stamp acknowledging receipt of the IDS and references. Applicants request that Examiner review the previously submitted IDS and references, and indicate such review by initialing the PTO-1449 form submitted with the IDS.

The Examiner objected to the Specification and Drawings for using the reference number "82" to refer to the disclosed "program" element. (Office Action, pg. 2) Applicants have amended the Specification to use the reference number "80" for the disclosed program. Applicants submit that these amendments to the Specification overcome the Examiner's objection to the Drawings and Specification.

1. Claims 1, 3, 5-7, 12, 14, 16-18, 23, 25, and 27-29 are Patentable Over the Cited Art

The Examiner rejected claims 1, 3, 5-7, 12, 14, 16-18, 23, 25, and 27-29 as anticipated (35 U.S.C. §102) by Lee (U.S. Patent No. 5,553,286). Applicants traverse this rejection for the following reasons.

Claims 1, 12, and 23 concern producing an executable file for execution by a computer, and require: receiving a plurality of programming language statements comprising a source program into the computer; translating the source program into an object module, wherein the object module is capable of including: a symbol reference; a symbol definition; attribute information for the symbol reference derived from the language statements; attribute information for the symbol definition derived from the language statements; binding object modules into a program object, wherein the attribute information is available when binding object modules into the program object; and resolving in the program object an external symbol reference in the object module with an external symbol definition in another object module.

The Examiner cited col. 6, line 51 to col. 7, line 1 and col. 8, lines 37-37 of Lee as disclosing the claim requirement that the object module include attribute information for the symbol reference derived from the language statements; and attribute information for the symbol definition derived from the language statements. (Office Action, pg. 3) Applicants traverse for the following reasons.

The Examiner found that the index numbers, size and offset fields discussed in cited cols. 6-7 disclosed the claimed attribute information for the symbol reference and definition. Applicants traverse, because the cited index, size and offset information concern limitations of a program object, which results in a program object constrained only by the amount of memory available. (Lee, col. 6, lines 51-61) The claims require that the object module include attribute information for the symbol references and definitions. The cited index, size and offset information is the structural information that defines the layout of the program object.

Further, the index, size, and offset information mentioned in the cited col. 8, lines 37-47 are used by the binder to overlay objects. The binder processes object modules, load modules, and program objects to create a single program object. (Lee, col. 7, lines 54-58)

Nowhere does the cited cols. 6-7 and 8 anywhere disclose that the object module include attribute information for the symbol definitions and references. Instead, the index, size, and offset information the Examiner cites concern the limitations of the program object and information used by the binder when constructing the single program object. Nowhere do these cited sections anywhere disclose that the cited index, size, and offset information comprise attribute information for symbol references and definitions in an object module as claimed.

The Examiner cited col. 8, lines 37-46 as disclosing the claim requirement of binding object modules into a program object, wherein the attribute information is available when binding object modules into the program object. (Office Action, pg. 3) Applicants traverse

The cited col. 8 discusses how the binder overlays items to form the single program object. Nowhere does the cited col. 8 anywhere disclose the claim requirement of making available attribute information for symbol references and definitions when binding object

modules. Instead, the cited col. 8 discusses how size, offset and sequence are used by the binder to overlay items when creating the single program object.

Accordingly, claims 1, 12, and 23 are patentable over the cited art because the cited Lee does not disclose all the claim requirements.

Claims 3, 5-7, 14, 16-18, 25, and 27-29 are patentable over the cited art because they depend from one of claims 1, 12, and 23, either directly or indirectly.

Claims 3, 14, and 25 depend from claims 1, 12, and 23 and further require that the object module is further capable of including fixed attribute information derived from language statements declaring attribute information for the symbol reference and symbol definition.

The Examiner cited col. 7, lines 4-30 and 47-49 and col. 9, lines 38-45 of Lee as disclosing the additional requirements of these claims. (Office action, pg. 3) Applicants traverse.

The cited col. 7, lines 4-30 discuss class attribute information that controls the loading and binding of items in the class, including physical organization, how the items are to be structured during binding, whether the class can contain address constants, whether the class should be loaded into storage, etc. The cited col. 7, lines 47-49 mentions that the binder logic is reduced by confining loading and binding variations to a finite set of class attributes.

Although the cited col. 7 discusses information used during binding and mentions examples of this information, nowhere does the cited col. 7 anywhere disclose that such information used during binding include fixed attribute information derived from language statements declaring attribute information for the symbol reference and definition. Nowhere does the cited col. 7 anywhere disclose that the attribute information comprises attribute information for the symbol reference and definition as claimed

The cited col. 9 mentions that the loader opens a file and that the object's structural data is read into memory. The loader allocates storage for each loadable segment in the object. The length, location and other characteristics of each block of storage will be determined by the class attributes. This cited section nowhere discloses that the attributes available to the binder include fixed attribute information derived from language statements declaring attribute information for the symbol reference and definition

Accordingly, claims 3, 14, and 25 provide additional grounds of patentability over the cited art because the cited Lee does not disclose the additional dependent claim requirements.

Claims 5, 16, and 27 depend from claims 1, 12, and 23 and require that the object module is further capable of including an address constant for a symbol referenced in the module and attribute information derived from language statements declaring attribute information for the address constant.

The Examiner cited col. 8, lines 48-52 as disclosing the claim requirement of attribute information derived from language statements declaring attribute information for the address constant, where the attribute information is for symbol definitions and references. (Office Action, pg. 3). Applicants traverse.

The cited col. 8 mentions that the binder stores the target address of a relocatable address constant and that class offsets are stored in the address constant. Although the cited Lee discusses address constants, nowhere does the cited col. 8 anywhere disclose the claim requirement that attribute information for symbol references and definitions is derived from language statements declaring attribute information for the address constant. Nowhere does the cited col. 8 anywhere disclose that an address constant provides attribute information for a symbol available during binding.

Accordingly, claims 5, 16, and 27 provide additional grounds of patentability over the cited art because the cited Lee does not disclose the additional dependent claim requirements.

Claims 6 and 7; 17 and 18; and 28 and 29 depend from claims 5, 16, and 27, respectively, and provide additional requirements concerning the requirement that address constants provide attribute information for the symbol references. Again, although the cited Lee discusses address constants, nowhere does Lee anywhere disclose that address constants provide attribute information for a symbol available during binding. For these reasons, claims 6, 7, 17, 18, 28, and 29 provide still further grounds of patentability over the cited art because they concern further requirements on the claimed address constant and symbol attribute information that is not disclosed in the cited Lee.

2. Claims 2, 4, 8, 9, 13, 15, 19, 20, 24, 26, 30, and 31 are Patentable Over the Cited Art

The Examiner rejected claims 2, 4, 8, 9, 13, 15, 19, 20, 24, 26, 30, and 31 as obvious (35 U.S.C. §103) over Lee in view of Fitzgerald (U.S. Patent No. 5,408,665). Applicants traverse for the following reasons.

Claims 2, 4, 8, 9, 13, 15, 19, 20, 24, 26, 30, and 31 are patentable over the cited combination because they depend either directly or indirectly from one of claims 1, 12, and 23, which are patentable over the cited art for the reasons discussed above.

Claims 2, 13, and 24 depend from claims 1, 12, and 23 and further require that the language statement is capable of indirectly declaring extended attribute information defined in another location in the object module. The Examiner cited the EXTDEF, EXTERN, and Module commands and col. 12, lines 5-33 of Fitzgerald as teaching the additional requirements of these claims. (Office action, pg. 5) Applicants traverse.

The cited col. 12 mentions that EXTDEF is an external name definition in the object module. Nowhere does the cited Fitzgerald anywhere teach or suggest the claim requirement that the language statement from which the symbol attribute information is derived is indirectly declared in another location in the object module. The cited figures in Fitzgerald show the relationship between external symbols (EXTRN) and public symbols in source and object modules. Nowhere in the cited Fitzgerald is there any teaching or suggestion of deriving attribute information for symbol references and definitions from a language statement indirectly declared in another location in the object module.

Accordingly, claims 2, 13, and 24 provide additional grounds of patentability over the cited art.

Claims 8, 19, and 30 depend from claims 1, 12, and 23 and further require the object module further includes an External Symbol Directory (ESD) including a record capable of indicating a symbol in the program, a location of the symbol in the program, and a pointer to attribute information in the program for the symbol.

The Examiner cited col. 15, lines 46-54 and FIG. 5b of Fitzgerald as teaching the claim requirement that the ESD include a pointer to attribute information in the program for the symbol. (Office Action, pgs. 5-6) Applicants traverse.

The cited col. 15 discusses a pointer to unresolved external symbols, and that if the pointer is unresolved then control proceeds to get the next external symbol from the Externals Table. Nowhere does the cited col. 15 of Fitzgerald anywhere teach or suggest the claim requirement of a pointer to attribute information for symbol definitions and references. Instead, the pointer in the cited col. 15 is for something entirely different – a pointer to an unresolved external symbol, and not a pointer to attribute information for symbol definitions and references as claimed.

Accordingly, claims 8, 19, and 30 provide additional grounds of patentability over the cited art.

Claims 9, 20, and 31 depend from claims 1, 12, and 23 and further require that the object module further includes a Relocation List Directory (RLD) including a record capable of indicating the location of the referenced symbol, a location of an address constant for the referenced symbol in the program, and a pointer to attribute information for the address constant in the program.

The Examiner cited the same col. 15 of Fitzgerald discussed above as teaching the requirement a pointer to attribute information for the address constant in the program. (Office Action, pg. 6) Applicants traverse.

As discussed, the cited col. 15 discusses a pointer to unresolved external symbols, and that if the pointer is unresolved then control proceeds to get the next external symbol from the Externals Table. Nowhere does the cited col. 15 of Fitzgerald anywhere teach or suggest the claim requirement of a pointer in the Relocation List Directory (RLD) to attribute information for the address constant in the program, where the claimed attribute information is for symbol references and definitions. Nowhere does the cited col. 15 anywhere disclose that the claimed pointer is in the RLD.

Accordingly, claims 9, 20, and 31 provide additional grounds of patentability over the cited art.

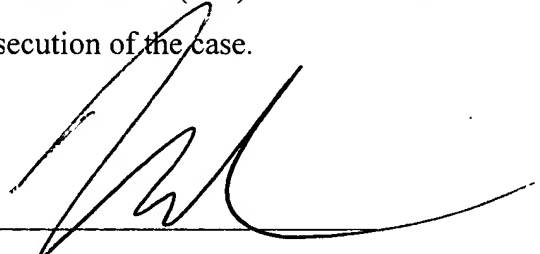
Conclusion

For all the above reasons, Applicant submits that the pending claims 1-33 are patentable over the art of record. Applicants have not added any claims. Nonetheless, should any additional fees be required, please charge Deposit Account No. 50-0585.

The attorney of record invites the Examiner to contact him at (310) 553-7977 if the Examiner believes such contact would advance the prosecution of the case.

Dated: April 8, 2003

By: _____


David W. Victor
Registration No. 39,867

Please direct all correspondences to:

David Victor
Konrad Raynes Victor & Mann, LLP
315 South Beverly Drive, Ste. 210
Beverly Hills, CA 90212
Tel: 310-553-7977
Fax: 310-556-7984